AUTHENTICATING SOFTWARE LICENSES

"EXPRESS MAIL" MAILING LABEL

NUMBER EL701911815US

DATE OF DEPOSIT October 30, 2001
I HEREBY CERTIFY THAT THIS PAPER OR FEE IS BEING DEPOSITED WITH THE UNITED STATES POSTAL SERVICE "EXPRESS MAIL POST OFFICE TO ADDRESSEE" SERVICE UNDER 37 CFR 1.10 ON THE DATE INDICATED ABOVE AND IS ADDRESSED TO THE COMMISSIONER OF PATENTS AND TRADEMARKS, WASHINGTON, D.C. 20231.

Ana R. Rivera
(TYPED OR PRINTED NAME OF PERSON MAILING PAPER OR FEE)

Ana R. Rivera
(SIGNATURE OF PERSON MAILING PAPER OR FEE)

Authenticating Software Licenses

Cross-Reference To Related Application

This application is related to and claims priority to U.S. provisional application entitled AUTHENTICATING SOFTWARE LICENSES having serial number 60/243718, by Raymond HO and Edward FUNG, filed October 30, 2000 and incorporated by reference herein.

Field of Invention

This invention relates to software licenses and in particular to authentication and enforcement of software licenses in computer systems.

Background of the Invention

Software in a computer system may be distributed in a number of ways. From the perspective of preventing unauthorized use, these distribution methods may be classified into three groups: unrestricted entitlement, restricted entitlement, and non-entitlement methods.

Unrestricted entitlement means that the software distributed with a computer system will run on any system for which it was designed, without any restrictions. Apart from the licensing and contractual agreement, there is nothing in the software to guard against unauthorized use. This method is not desirable for expensive software.

Restricted entitlement means that the software contains some means to limit itself to run only on the computer system for which it is authorized. A common restriction method is to encode hardware specific information in the computer system so that the software can verify the information at system start-up. Another method is to make the software unique for every computer system.

This entails unique compilation of the software for each distribution, which is a very costly operation.

Non-entitlement means the software is disabled when distributed, and requires a separate authorization method to enable the software. This method is commonly adopted in systems where a single generic distribution medium is used to distribute all of the software, and software components or packages within may be enabled or disabled according to license contract.

It is widely believed to be very difficult, if not impossible, to design a software protection method that cannot be defeated over the long run. It would be advantageous to devise a protection method that would reduce the incentive for a potential counterfeiter to counterfeit the system, thereby achieving the objective of protecting the software from software piracy.

The problem of software piracy is acute with a particular class of computer systems: Internet Appliances. An Internet Appliance is generally a computer system that performs some predetermined functions while connected to the Internet. The Internet Appliances typically consist of computer hardware with embedded software. The hardware includes a storage medium and a network interface card.

Software embedded in an Internet Appliance tends to be compact. It is not uncommon to store the entire system software in a storage medium that has only a few megabytes of capacity. This type of storage medium is usually small and very portable (such as CompactFlash and SIM cards). Because of wide adaptation and portability of such media, digital content inside such mediums can be illegally duplicated very easily.

It is therefore an aspect of an object of the present invention to provide a method and an apparatus for protecting the embedded software in computer systems, such as Internet Appliances, against unauthorized use, while being relatively cost-effective to deploy.

An aspect of another object of this invention is to provide a method and an apparatus for protecting the software in a computer system, while allowing a user to make legitimate backup copies of the software.

An aspect of another object of this invention is to provide a method and an apparatus for a user to quickly re-entitle the software in a computer system in the event that the software enters a non-entitlement state.

Summary of the Invention

According to an aspect of the present invention, there is provided a method and an apparatus for using an encrypted unique digital signature ("engraved signature") as a uniquely definable signature to control the use or execution of software in a computer system. The computer system has a Network Interface Card ("NIC") with a Media Access Control ("MAC") address. On start up, the engraved signature is retrieved from the persistent storage medium of the computer system and the MAC address is retrieved from the NIC. The MAC address is unique according to industry standards and therefore uniquely identifies the NIC being used in the computer system. A computed encrypted signature is generated using the MAC address. Where the computed encrypted signature does not match the engraved signature, the execution of the software is halted.

According to another aspect of the present invention, the MAC address is encrypted using a one-way encryption algorithm where the encryption key used to perform the encryption is different from the decryption key used to perform the decryption.

According to a further aspect of the present invention, there is provided a method of storing an engraved signature into a persistent storage medium by initializing said medium with a blank signature, preferably during the software reproduction process. The blank signature is a unique predefined pattern of binary code. During system startup, the software protection program checks to determine if the signature in said medium is blank or not. If blank, the protection software computes an encrypted signature based on the MAC address of the NIC in the computer system. The computed encrypted signature is stored in the persistent storage medium as the engraved signature for future authentication. Preferably, this process of engraving the signature is done once at the premises of a manufacturer before the computer system is shipped to the user.

Thus, users may back up the protected software without restriction as the engraved signature restricts the copies of the software from being used in unauthorized computer systems.

Brief Description of the Drawings

In the accompanying drawings:

Figure 1 is a block diagram of a software protection program having an engraved signature for protecting software of a computer system according to an embodiment of the present invention;

Figure 2 is a flowchart of the steps of generating an encrypted signature for the computer system of Figure 1;

Figure 3 is a flowchart of the steps to authenticate the computer system for a license to the software according to the software protection program of Figure 1;

Figure 4 is a flowchart of the steps to set up the software protection program with the engraved signature of Figure 1; and

Figure 5 is a flowchart of the steps to automatically set up the software protection program with the engraved signature of Figure 1.

Detailed Description of the Preferred Embodiments

Referring to Figure 1, there is shown a block diagram of a software protection program for a computer system 10 according to an embodiment of the present invention. The computer system 10 comprises a central processing unit (CPU) 12, a random access memory (RAM) module 14, network interface card (NIC) 16 embedded with a unique Media Access Control (MAC) address 18 that can be read electronically, and a persistent storage medium 20. The NIC 16 can be an external adaptor card or part of an onboard chip set. The persistent storage medium 20 contains the system software for the computer system 10, plus software 28 protected by the software protection program. The software protection program comprises a signature engraving program 22, a signature authentication program 24 and an engraved signature 26.

The engraved signature 26, which is a 128-bit binary code, is stored in the persistent medium as a 32-byte hexadecimal character string where every byte (8 bit) of the signature is represented by 2 hexadecimal characters. The initial digital code of the signature 26 is blank. A blank signature 26 is a predefined code pattern, the value of which is arbitrarily defined, but which value should not be the same as a signature computed from a MAC address.

The MAC address 18 embedded in the NIC 16 is a unique hardware identifier specified by the NIC hardware manufacturer. MAC addresses on all NICs are unique as per industry standard. The MAC address 18 is a 48-bit binary code created and encoded by the NIC manufacturer and is readable by the software running in the computer system 10. A computed encrypted

signature is generated based on the MAC address 18 of the computer system 10. The engraved signature 26 is an encrypted signature based on one authorized MAC address.

The signature authentication program 24 authenticates the computed encrypted signature by comparing it with the engraved signature 26. The software 28 is authorized or authenticated where the computed encrypted signature matches the engraved signature 26. The program 24 is preferably executed during the system start-up so that unauthorized use of the software 28 is detected as soon as possible, but the program 24 may also be executed at any time when the computer system 10 is running.

The engraved signature 26 is fabricated using unique hardware identification of the MAC address 18 by means of encryption. An encryption method is implemented using a publicly available algorithm called Block Cipher SQUARE. The algorithm used is adopted from a published research paper by Joan Daemen, Lars Knudsen, and Vincent Rijmen. entitled "The Block Cipher Square", Eli Biham, editor, Fast Software Encryption '97, volume 1267 of Lecture Notes in Computer Science, pages 149--165, Haifa, Israel, January 1997, Springer-Verlag.

The algorithm is a one-way encryption method where the encryption key used to perform encryption is different from a key used to perform decryption. Only the encryption method is required and used in accordance with this invention. It will be understood by those skilled in the art that other encryption methods may also be used without departing from the scope of this invention.

The encryption method encodes and decodes 128-bit binary numbers. The encryption method is a 2-step process in which an encryption key is generated first and is used by the second step to create the encrypted data. The MAC address 18 is only a 48-bit code. The rest of the 80-bit code is arbitrarily

assigned to complete the 128-bit code input required by the encryption method. The 80-bit code is hard coded into the software protection program.

Referring to Figure 2, there is shown a flowchart of the steps for generating an encrypted signature for the computer system 10 of Figure 1. At step 200, the MAC address 18 is read then, at step 202, the 48-bit MAC address 18 is combined with the 80-bit code for a unique hardware ID. An encryption key is created using the unique hardware ID by the key generation ("KeyGen") logic component of the software protection program (step 204). The encrypted signature is then created from the encryption of the unique hardware ID using the encryption key (step 206). Thus, the encrypted signature is the computed encrypted signature for authentication purposes to the signature authentication program 24, and is the engraved signature 26 when the encrypted signature is created for the signature engraving program 22.

Referring to Figure 3, there is shown a flowchart of the steps to authenticate the computer system 10 for a license to the software 28 according to the software protection program of Figure 1. At step 298, the signature authentication program 24 is started by the execution of the software 28. At step 300, the engraved signature 26 is read from the persistent storage medium 20 and stored in RAM 14 for use by later steps. At step 302, the MAC address 18 is read from the Network Interface Card 16 and then, step 304, the computed encrypted signature is generated by encrypting the MAC address 18. At step 306, the computed encrypted signature is compared to the engraved signature 26. If No, the computed encrypted signature does not match with the engraved signature 26, then the execution of the software 28 is halted (step 308). If Yes, the computed encrypted signature matches the engraved signature 26, then the execution of the software 28 continues (step 310).

Where the software 28 is the operating system of the computer system 10, the operation of the computer system 10 is thus halted on boot up if the computed encrypted signature does not match the engraved signature 26.

Referring to Figure 4, there is shown a flowchart of the steps to set up the software protection program with the engraved signature 26 of Figure 1. At step 400, the MAC address 18 is read from the Network Interface Card 16 and, step 402, display the MAC address 18 to a user. The user then contacts the licensor of the software 28, provides the MAC address 18, and obtains a signature there from (step 404). The licensor uses the MAC address 18 to generate the computed encrypted signature for the user. The signature from the licensor is then saved as the engraved signature 26 (step 406).

The steps of Figure 4 may be used as a non-entitlement means for enabling the software 28. Further, if for any reasons the engraved signature 26 in the persistent storage medium 20 becomes corrupted, then the steps of Figure 4 may also be used to re-setup the engraved signature 26.

Referring to Figure 5, there is shown a flowchart of the steps to automatically set up the software protection program with the engraved signature 26 of Figure 1. At step 500, the signature authentication program 24 is started by the execution of the software 28. At step 502, the engraved signature 26 is read from the persistent storage medium 20 and stored in RAM 14 for use by later steps. At step 504, the MAC address 18 is read from the Network Interface Card 16 and then, step 508, the computed encrypted signature is generated by encrypting the MAC address 18. At step 510, the engraved signature 26 is compared to determine if it is a blank signature. If Yes, the engraved signature 26 matches the blank signature, then, step 512, the signature engraving program 22 engraves or stores the computed encrypted signature in the persistent storage medium 20 as the engraved signature 26. The execution of the software 28 continues (step 514). At step 512, the software protection program may disable or erase the signature engraving program 22 after one engraving for greater security.

If at step 510, the engraved signature 26 does not match the blank signature, then, step 516, the computed encrypted signature is compared to the engraved signature 26. If No, the computed encrypted signature does not match with the engraved signature 26, then the execution of the software 28 is halted (step 518). If Yes, the computed encrypted signature matches the engraved signature 26, then the execution of the software 28 continues (step 514).

When the software protection program is run for the first time after the software protection program is reproduced from a master copy, the engraved signature 26 has the blank signature. Thus, this process of engraving a signature is preferably done by the computer system manufacturer during system integration, but it can also be done at other times and by other parties.

The computed signature that is stored as the engraved signature may further be encrypted using another one-way encryption method. In this embodiment, the computed signature is encrypted using an encryption key of said another one-way encryption method by, for example, the manufacturer of the computer system during system integration. The signature authentication program only needs a decrypting key to read the engraved signature. In this manner, greater security can be achieved as the encryption key of said another one-way encryption method is not otherwise on the computer system.

It will be understood by those skilled in the art that other signatures in the computer system can be used for the same purpose, as identifiers, whether unique or mostly unique to the particular computer systems. The other signatures include the serial number of CPUs, hard drive format code numbers, code number of computer system "add-ons", or a combination of these signatures to form unique signatures. Mostly unique means that the identifier used is sufficient for authentication purposes even though it is not unique for some computer systems.

Although preferred embodiments of the invention have been described herein, it will be understood by those skilled in the art that variations may be made thereto without departing from the spirit of the invention or the scope of the appended claims.